

### Claims

1. A method in a network of the delivery of files from a server computer to a client computer, characterized in that the client computer is polling the server computer  
5 looking to see in the server if at least one new file associated with this client computer is available and wherein the client computer is downloading the files associated with this client computer at specific intervals.
2. The method as claimed in claim 1, wherein the client computer is all by itself  
10 polling the server computer at specific intervals and wherein the client computer itself controls the downloading of the updates which occurs at times during which the downloading does not disturb the working of the client computer.
3. The method as claimed in claim 1, wherein a specific manifest for each client  
15 computer is created in the server computer, wherein a model version of this manifest is maintained in the server computer, wherein the respective client computer downloads its model version of the manifest to the client computer upon the client computer recognized a model version change and wherein the files are downloaded, activated, deactivated and deleted in the client computer at specific  
20 times.
4. The method as claimed in claim 3, wherein to deliver a new application or file to a client, an administrator computer adds file records representing the new application, i.e. a new fileset, into the database of the server computer and  
25 associates these file records to a specific client computer, wherein the server computer calculates a new manifest for each client computer.

5. The method as claimed in claim 1, wherein the client computer compares the downloaded model version of the manifest to the manifest already present in the client computer, wherein the client computer computes a delta on each fileset list under management upon downloading the manifest, so that difference between the  
5 newly received and already present manifest is established in the client computer and wherein appropriate local operations in the computer are performed immediately based on said delta, so that the client computer downloads and activates or deletes files immediately based on the delta of the fileset list, and wherein the client computer reports back to the server computer the receipt of the  
10 update and the changes carried out in the client computer, so that the state of the files on the client computer or computers is automatically transmitted to and maintained in the server computer.
6. The method as claimed in claim 5, wherein the client computer stores the  
15 scheduled actions in a cache file, wherein this cache file contains a schedule of actions to perform as well as the status of each action and wherein the client computer executes the scheduled actions at the time specified by the client's schedule section of its cache.
- 20 7. The method as claimed in claim 6, wherein the client computer checks the state of each fileset in its cache at a specified time interval by comparing the state returned from the file system of the operating system to the state stored in the cache.
- 25 8. The method as claimed in claim 7, wherein the checksum of each file is read from the fileset list and wherein the file is downloaded and activated thereby maintaining a steady state of all the filesets in the cache, if the checksum

contained in the fileset list is different from the checksum computed in real time from the operating system.

5 9. The method as claimed in claim 8, wherein the format of the files saved to the local disk in the client computer contains all the information needed to copy the files to the server so that the files will contain the proper information when they are delivered to the client.

10 10. The method as claimed in claim 5, wherein upon downloading the manifest the client computer is reading its manifest and wherein an entire fileset is scheduled for download at the time specified in the manifest, if the fileset did not exist in the previous manifest.

15 11. The method as claimed in claim 5, wherein the respective client computer has a preference that directs the client computer to check in with the server computer for a change in the model version on the server computer at a specific time interval of an interactive session and wherein the client computer will retry checking in with the server computer its specific manifest from the model version at the next specific time interval, if the server computer is not available when the client  
20 computer checks in.

25 12. The method as claimed in claim 5, wherein the status of the action is sent back to the server computer, where the administrator can view the status of this action and wherein the administrator computer creates reports on the status of multiple clients.

13. The method as claimed in claim 12, wherein the administrator computer creates a snapshot of a hard disk in the administrator computer, wherein a snapshot catalog of all the files on the disk in the administrator computer stores the files in a list written to this local disk, wherein the administrator computer alters the content of the hard disk in the administrator computer by running an installer or copying files to the disk, wherein a second snapshot is taken, wherein a delta of the first snapshot and the second snapshot is displayed to the administrator computer and wherein the administrator computer then chooses the files added or modified that are to be included in a fileset associated with a specific client computer.

14. The method as claimed in claim 13, wherein the delta of the snapshot can be saved to the server computer or to the local disk in the administrator computer.

15. The method as claimed in claim 5, wherein the client computer process is monitored in such a way that it reports the real-time status of the client computer from the administrator interface, wherein the connection to the client process is through a TCP/IP socket and wherein the information about the status is encapsulated with the TCP/IP packets.

16. The method as claimed in claim 5, wherein the network is designed in such a way that it in no way adversely effects the client computers or the server computer, if the client computer or server computer being off.